



QuickTime for the Web

Session 606





QuickTime for the Web

Steven Gulie
Senior Technical Writer
Multimedia Web Monkey

Introduction

- QuickTime for Web delivery
 - Different parts of QuickTime
 - Plug-in, ActiveX control, QT Player, QT file format
 - HTML
 - Tags and extensions
 - Scripting
 - Client-side (JavaScript)
 - Server-side (CGI/SSI)



What You Will Learn

- Tags and Extensions
 - Detecting QuickTime, launching QT Player, going full-screen
- Plug-in vs Player
- QT file format and the Web
- Creating Web movies dynamically
- Scripting the plug-in and ActiveX
- Wired actions and server-side scripts



HTML Tags

- Embed Tag
 - Netscape-style plug-ins
 - Introduced to handle new media
 - Designed to handle 1 MIME type
 - SRC, TYPE, PLUGINSPAGE
 - Passes other parameters to plug-in



Embed Tag

< EMBED

HEIGHT=yy WIDTH=xx

SRC="server.domain/path/mediafile.mov"

TYPE="video/quicktime"

PLUGINSOURCE=

<http://www.apple.com/quicktime/download>

OTHER="stuff" >



Embed Tag

- Depends on MIME type
 - QuickTime parameter: QTSRC
 - Prevent "**hijacking**"
 - Play other media
 - MP3, SMIL, RTSP://



Embed Tag

- Embed Tag works with
 - Plug-in, ActiveX control
 - All current browsers
 - Mac OS and Windows
 - One problem:
 - IE/Win 5.5+ ignores Pluginspage
 - If you need QT or ActiveX, no help



Object Tag

- Originally for Plug-ins, ActiveX, other
 - W3C standard
 - Duplicates EMBED functionality
 - MicroSoft extensions
 - Unique Class ID, Code base
 - ActiveX control only
 - IE 4 and later for Windows only



Object Tag

< OBJECT

HEIGHT=yy WIDTH=xx

CLASSID=

"clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"

CODEBASE=

<http://www.apple.com/qtactivex/qtplugin.cab> >

< param

name="SRC"

value="server.domain/path/mediafile.mov" >

< param name="OTHER" value="stuff" >

</OBJECT>



Object and Embed Tags

- Embed Tag
 - IE/Win 5.5+ ignores Pluginspage
 - Subject to hijacking (IE/Win)
- Object Tag
 - Prevents hijacking in IE/Win
 - Works only with IE/Win
- Use both



Object and Embed Tags

```
< OBJECT HEIGHT=yy WIDTH=xx  
  CLASSID=  
    "clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"  
  CODEBASE=  
    http://www.apple.com/qtactivex/qtplugin.cab >
```

```
< param name="SRC" value="mediafile.mov" >
```

```
< EMBED HEIGHT=yy WIDTH=xx  
  SRC="mediafile.mov" TYPE="video/quicktime"  
  PLUGINSOURCE=  
    http://www.apple.com/quicktime/download  
/> </EMBED>
```

```
</OBJECT>
```



Gateway Concept

- Once the user has QuickTime ActiveX, **<embed>** tag works fine
- Surround the **<embed>** tag on your **entry page** with the **<object>** tag
- Subsequent pages can use **<embed>** tag alone
- Problem: User may not come through gateway



Applications Developers

- If your app generates HTML, create both **<object>** and **<embed>** tags by default.
 - You may want to allow the option to use **<embed>** only, if a **script** is used to detect IE/Win and QT/ActiveX
 - This prevents **<object>** tag from offering to download QuickTime
- Okay to use **NAME=src VALUE=actual** instead of **SRC=dummy QTSRC=actual**
 - Unless **QTSRCDONTUSEBROWSER** is used



QuickTime URLs

- Of course HTTP://url, but also
 - FTP, RTSP/RTP
 - File://path/filename
 - QT Player/browser problem in Mac OS X
 - Data:
 - Base64
 - JavaScript:
 - "**<url> T<target> E<embed tags>**"



Little-Known Parameters

SaveEmbedTags

AllowEmbedTagOverrides

AutoPlay="@hh:mm:ss:ff"

MovieName

Pan, Tilt, FOV, Node, Hotspotn

EnableJavaScript

URLSubstitute="<string>:<sub>"



Plug-In or Player?

- Plug-in (or ActiveX control)
 - Understands Embed parameters
 - Reads settings in **'plug'** atom
 - Viewable in Plug-in Helper
 - Handles HREF tracks with **A<url >**
 - Uses the browser for downloads
 - Multiple open movies



Plug-In or Player?

- QuickTime Player
 - Uses settings from Movie menu
 - Stores settings as Movie User Data, **'udat'** atom inside **'moov'** atom
 - Ignores some things:
 - **A<url>** in **nonstreaming** HREF tracks
 - **E<>** extension to URL
 - Handles non-html URLs natively
 - Has full-screen modes



QuickTime Player

- Launching from a Web page
 - Poster movie
 - HREF = “<url> T<quicktimeplayer>”
 - .qtl file
 - Type = **application/x-quicktimeplayer**

```
<?xml version="1.0"?>
```

```
<?quicktime type="application/x-quicktime-media-link"?>
```

```
<embed src=http://server.path/movie.mov />
```



Full-Screen Movies

- Full-screen modes
 - Normal, Double, Half, Full, Current
 - Set mode using
 - API, AppleScript (**'ptv'** atom)
 - Wired sprite action **beginFullScreen**
 - SMIL **qt:full-screen="full"**
 - XML (.qti or .mov)
<embed src=<http://server.path/movie.mov>

fullscreen="full" />



Scripting HTML Tags

- Check for QuickTime, browser, OS
- Write object/embed tags or:
 - Skip tags (background audio)
 - Use alternate inline content (jpeg)
 - Redirect to alternate content page
 - Redirect to download page
 - Your “You need QuickTime” page
 - www.apple.com/quicktime/download



Detecting QuickTime

- Two ways to detect QuickTime:
 - JavaScript/VBScript
 - Get browser type, version, OS, then check for QuickTime:
 - Netscape / Mozilla / IE5 for Mac
navigator.plugins
 - IE for Windows
**CreateObject("QuickTimeCheckObject
.QuickTimeCheck.1")
theObject.IsQuickTimeAvailable(0)**
 - Use QuickTime to detect itself



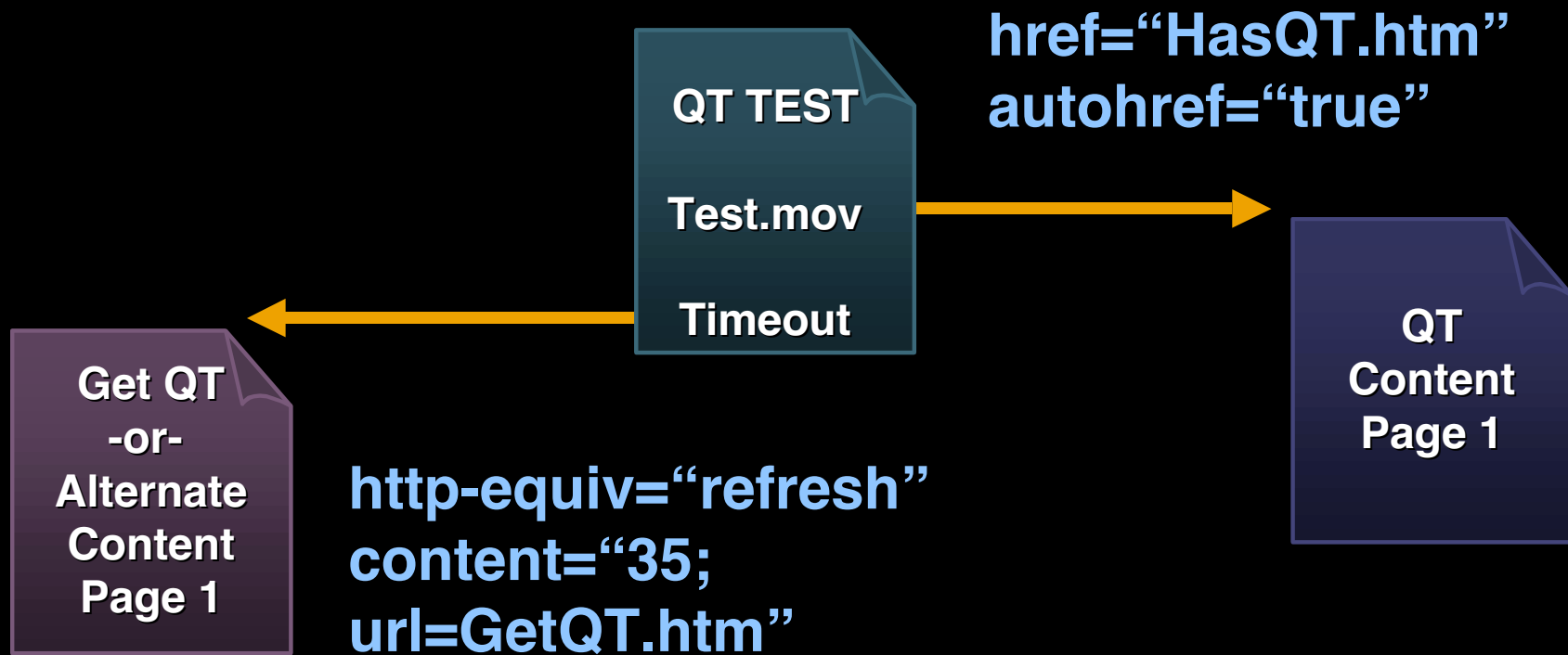
Detecting QuickTime

- Use QuickTime to detect itself
 - QT movie redirects to new page
 - href="HasQT.htm**
 - autohref="true"**
 - HTML redirects to alternate page on timeout if QT not installed

HTML **http-equiv="refresh"**
content="35;url=GetQT.htm"



Detecting QuickTime



Detecting QuickTime

- JavaScript/VBScript
 - Pro
 - Can prevent prompt for download
 - Only write tags if QuickTime present
 - Con
 - Not 100% reliable
 - False positive; MIME type mis-config
 - False negative; old QT, old IE/Mac



Detecting QuickTime

- Detecting QuickTime with QuickTime
 - Pro:
 - 100% reliable
 - Will prompt for download if needed
 - **Pluginspage** or **ClassID**
 - Con:
 - Can not prevent download prompt if QuickTime is not installed



QuickTime File Format and the Web

- Fast start
- Slow start
- Flattening
- Reference movies
- Storing data in external files
- Copy protection



QuickTime File Format

- Movie meta-data (movie resource)
 - Track descriptions
 - Type, duration, location of sample data
 - Alias, URL, “This file”

‘moov’ atom

- Sample data
 - May not be in same file

‘mdat’ atom (if in a movie file)



QuickTime File Format

- Fast Start
 - Progressive download
 - Movie meta-data stored first
 - **'moov'** atom (movie resource, movie)
 - **'mdat'** atom (sample data)
 - Dumpster, HexEdit
 - **FlattenMovie, FlattenMovieData**



QuickTime File Format

- Slow Start
 - Downloads before playing
 - Sample data stored first
 - **'mdat'** atom (sample data)
 - **'moov'** atom (movie resource, movie)
 - Update movie resource
 - Save without flattening



QuickTime File Format

- Flattening

FlattenMovie, FlattenMovieData

- Makes movie file self-contained
- Puts movie resource in data fork
- Stores **'moov'** atom before **'mdat'**
- Resolves all references!!!
- Interleaves sample data
- Puts **"preload"** data up front



QuickTime File Format

- Reference movies
- Movie files with external sample data
 - True reference movies
 - Refer to other movie files
 - Select movie best suited to viewer
 - Pointer movies
 - URL or file pointer
 - Movies with external dependencies



QuickTime File Format

- True reference movies

- 'rmra' atom

- Points to 1 or more movies or streams
 - May **not** have 'moov' atom
 - Flatten default movie into file
 - Import from XML
 - XMLtoRefMovie, API, QTPlayer



QuickTime File Format

- True reference movies
 - Contain selectors as well as refs
 - Many selectors available, including
 - Connect speed
 - CPU, OS, System Language
 - Screen resolution
 - QuickTime version
 - Presence of specific component



QuickTime File Format

- Pointer movies
 - Contain a single URL or file ref
 - May not have a **'moov'** atom
 - XML files with **.qtl** or **.mov** extension
 - RTSPtext RTSP://server/path/movie
 - Create using text editor or script
 - Contain only a streaming track
 - Open stream in QT Player, Save



QuickTime File Format

- Movies with external dependencies
 - Do contain a movie resource
 - May contain some movie data
 - **'mdat'** atom
 - Contain **URL** or **alias data** ref(s)
 - Alias: open movie from file
 - URL: open movie from URL
 - Save to new file, do not flatten



QuickTime File Format

- Copy protecting movie files
 - Jujitsu
 - Make copying work for you
 - Polite
 - KIOSKMODE
 - Media
 - Media keys, QuickTime restrictions (disallow saving), any encryption



QuickTime File Format

- Copy protecting movie files
 - Technically difficult
 - Example: Streaming movies
 - Not encrypted; anyone in this room
 - Can not “get” from server or drag from browser cache
 - Too complicated for 99.9% (99.99?)
 - Too much work and no challenge for hackers

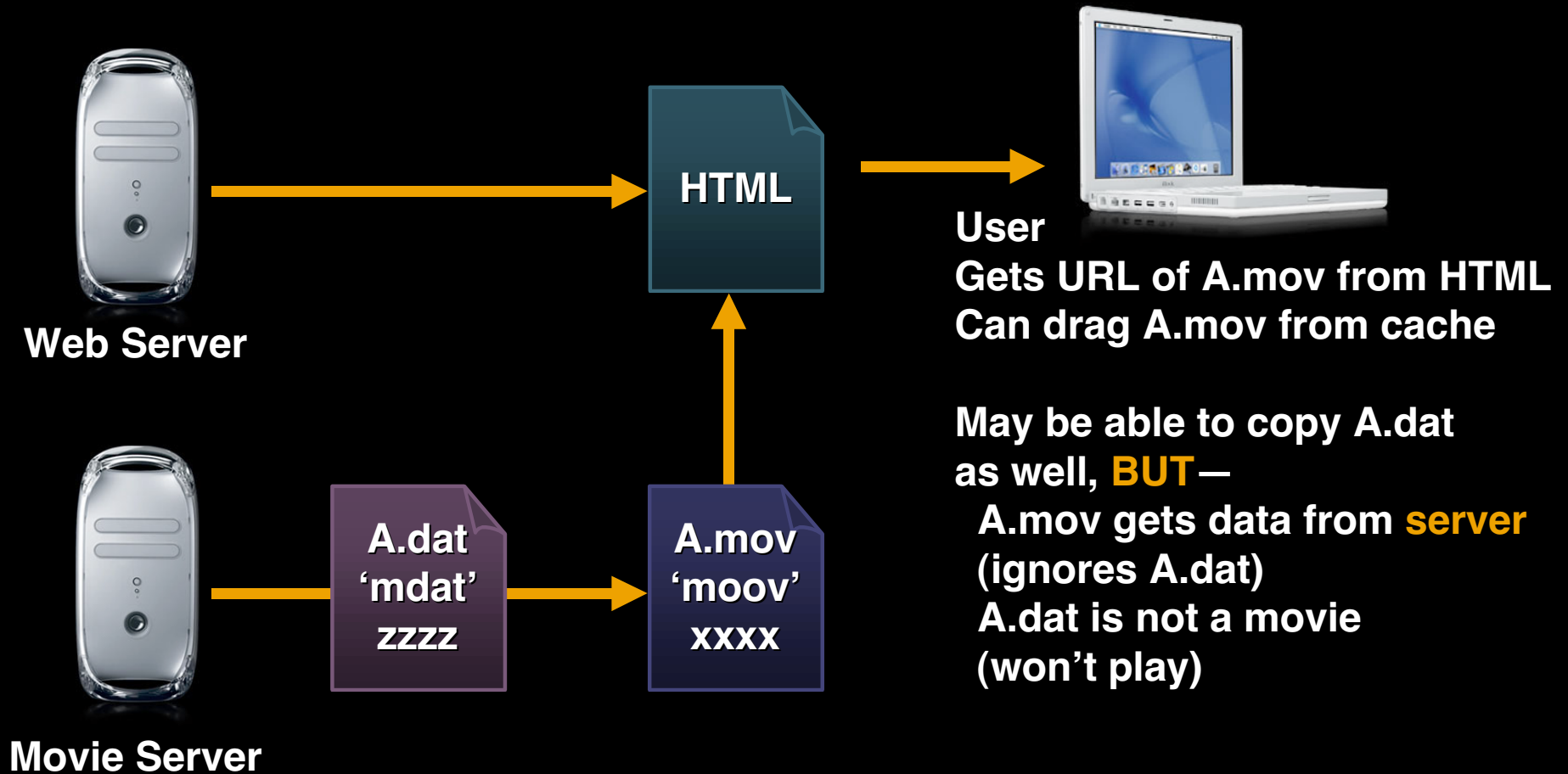


“Box of Rocks” Defense

- Goal: Make fast-start roughly as secure as streaming
 - How?
 - URL data references in movie file
 - Movie file always gets data from server
 - Compressed movie header
 - URL of data file(s) not in plain text
 - External data in non-movie file(s)
 - A data file is not a playable movie



“Box of Rocks” Defense



“Box of Rocks” Recipe

- Make slow-start movie
 - Open movie, save as **.dat** file
 - Movie resource at end of file
- Put slow-start movie on server
- Open URL, save as new file
 - Do not flatten, compress headers
- Delete **‘moov’** atom from **.dat** file



Dynamic Web Movies

- Use text to create movies or customize the viewer experience
 - SSI, CGI, JavaScript
 - SMIL
 - .qtl files
 - Text atom hacks
 - XML import



Dynamic Web Movies

- SMIL
 - Text: made by any CGI/SSI script
 - Use any available media
 - Mix and composite audio, stills, video, movies, text, streams
 - Generate text tracks dynamically
 - Create on the fly in response to:
 - Submitted form, cookie, wired action, quasi-random (movie of the day)



Dynamic Web Movies

- SMIL example
 - You have a database
 - MP3, video, CD item, CD artwork
 - User selects song list (JavaScript)
 - CGI generates SMIL text file
 - `<seq>`
 - `<par>` mp3, video or CD artwork, HREF to CD
 - `<par>` mp3, video or CD artwork, HREF to CD
 - `</seq>`



Dynamic Web Movies

- XML (saved as **.qtl** or **.mov** files)
 - Text: made by any CGI/SSI script
 - Use any available movie(s)
 - Launch in QuickTime Player
 - Full-screen movies
 - Create on the fly in response to:
 - Submitted form, cookie, wired action, quasi-random (movie of the day)



Dynamic Web Movies

- XML files
 - MIME type = application/quicktimeplayer
 - XML syntax specifies src, params

```
<?xml version="1.0"?>
```

```
<?quicktime type="application/x-quicktime-media-link"?>
```

```
<embed src=http://server.path/movie.mov  
        fullscreen="full" qtnext="2nd.mov"/>
```

- Save as **.mov** file to use Plug-in
- Save as **.qtl** file to use QuickTime Player



Dynamic Web Movies

- Text atom hacks
 - Text file with **.mov** suffix
 - First eight chars specify importer
 - TEXTtext Text, formatting {optional}
 - RTSPtext RTSP://MovieURL
 - SMILtext **<smil>**
 - XML file (not really a TA hack) save with **.qtl** or **.mov** file extension



Scripting the Plug-In

- JavaScript
- Netscape/Mozilla
 - Plug-in is fully scriptable
- IE for Windows
 - ActiveX is fully scriptable in QT6
- IE for Macintosh
 - Up to Microsoft



Scripting the Plug-In

- JavaScript

- Start, stop, go to time
- Enable and disable tracks
- Change volume, layer, matrix
- NLE

- QuickTime URLs

**HREF="Javascript: function{'params'}"
HREF, GoToURL, HOTSPOT, HREFTrack**



Wired Actions and Scripts

- GoToURL
 - Trigger CGI/SSI
 - Server/path/my.cgi?data
- Exchange Lists (XML)
 - Send and receive XML files
 - Movie can report user input
 - Movie wired actions based on XML
- Open child movie



QuickTime Roadmap

600 The State of QuickTime in 2002

Room A2
Wed., 9:00am

601 Building QuickTime Savvy Apps

Room A2
Wed., 10:30am

602 QuickTime for Video-Intensive Applications

Room A2
Wed., 2:00pm

603 Media Integration With QuickTime

Room A2
Wed., 3:30pm

604 Delivering Content via Interactive QuickTime

Room A2
Wed., 5:00pm



QuickTime Roadmap

FF010 QuickTime:

Room J1
Fri., 10:30am

606 QuickTime for the Web

Room A2
Fri., 2:00pm

**607 QuickTime and MPEG4:
A Technical Overview**

Room A2
Fri., 3:30pm



Who to Contact

Steven Gulie (QuickTime for the Web)

QuickTime Tech Writer

gulie@apple.com

Eric Carlson (Plug-in, ActiveX)

QuickTime Software Engineer

ericcarlson@apple.com



For More Information

- “QuickTime for the Web”
<http://developer.apple.com/techpubs/quicktime/qtdevdocs/QT4WebPage/>
- QuickTime Developer Documentation
<http://developer.apple.com/techpubs/quicktime/qtdevdocs/>
- Tutorials, Sample Code, Tools, SDK
<http://developer.apple.com/quicktime/>
- Other places
 - www.codeccentral.com
 - www.blueabuse.com
 - Mailing list: QuickTime-Talk lists.apple.com
 - Mailing list: QuickTime-API lists.apple.com



Technical Documentation

- All QuickTime tech docs:
<http://developer.apple.com/techpubs/quicktime/qtdevdocs/>
 - Note especially the links for:
 - PDF (includes drafts)
 - Function Index (complete API)
 - Site Updates
 - What's New in QuickTime (5, 4)
 - Tech Notes, Sample Code, Ice Floe
- OS X Help
 - Help Center > Developer > QT



Reminder

The QuickTime Engineering Team
Is Holding a “Hands On Lab” Everyday
from 1:00-4:00pm in Room G . . Stop By!



 **WWDC2002**

 **WWDC2002**

 **WWDC2002**